Administrar Cuestionario

Para mantener el cuestionario del siglo xix, hemos decido crear un administrador, que nos permita crear nuevas preguntas, modificar la preguntas que ya hay en nuestra base de datos y eliminarlas si es necesario. Para acceder es necesario iniciar sesión en wordpress ya que es de contenido privado. Una vez dentro podremos acceder a la url:

http://iescelia.org/sigloxix/administrar-cuestionario/

**Componentes:**

*Almacenamiento:*

Se opera sobre la misma tabla que el cuestionario.

- wp\_preguntas -> Esta guardara las preguntas.

Tiene los siguientes campos:

- id (int)

- enunciado (varchar)

- respuesta1 (varchar)

- respuesta2 (varchar)

- respuesta3 (varchar)

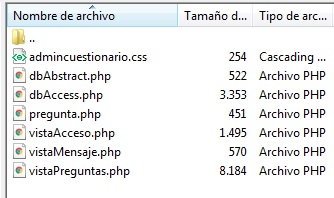
- dificultad (varchar) que se almacenan las preguntas fáciles, normales y difícil. Valores:(facil, normal, dificil).

- imagen (varchar)

*Archivos:*

A partir de la plantilla usada de wordpress alojada en wp-content/themas/honma/admincuestionario.php, se ha incluido el código correspondiente para relacionar los demás archivos creados en la carpeta admincuestionario. Los archivos se encuentran en:

wp-content/themas/honma/admincuestionario.



**Funcionamiento:**

*Resumen:*

El administrador del cuestionario comienza con una pantalla de bienvenida en la que podremos elegir las preguntas que deseamos gestionar según su dificultad.



Cuando pulsemos en “Empezar” en la siguiente pantalla se cargaran las preguntas de la dificultad escogida y además en la parte superior un botón para insertar una pregunta nueva y otro para volver a la pantalla anterior y seleccionar otra dificultad.

- Insertar

Al pulsar se muestra o se oculta el formulario para insertar una pregunta nueva.



- Volver a selección de dificultad

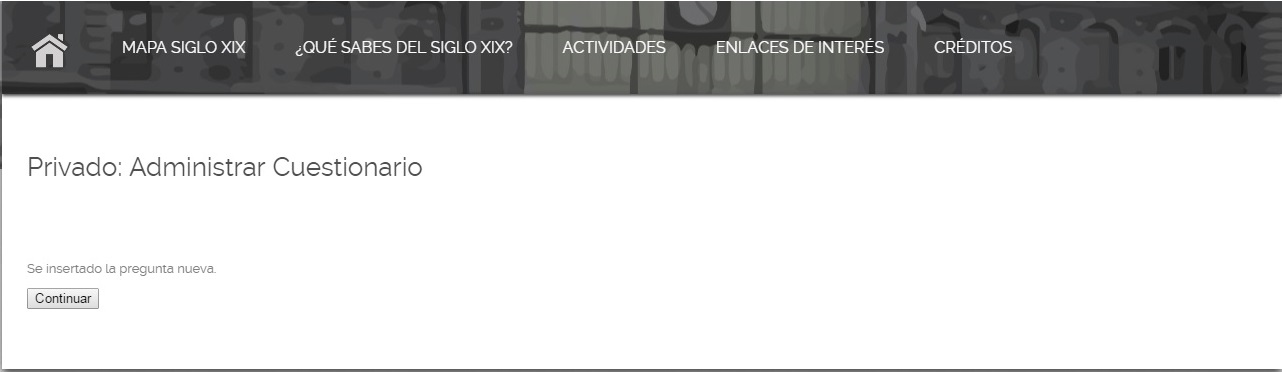
Al pulsar volveremos a la pantalla de bienvenida.

- Preguntas



Cada pregunta se carga en una capa diferente con un formulario en el que tenemos las opciones borrar (Saldrá un mensaje para confirmar) y editar (hará que los campos de texto sean editables y aparezcan dos botones, confirmar y cancelar).

Al realizar cualquier operación ya sea insertar, editar o borrar se mostrará una pantalla con el mensaje y un botón para continuar que nos llevará de nuevo a las preguntas que estábamos gestionando.



*Patrón de arquitectura:*

El modelo usado ha sido MVC (modelo vista controlador).

*Flujo de ejecución*

Cada vez que se recarga la página o se realiza una acción en ella recargará la página (http://iescelia.org/sigloxix/administrar-cuestionario/)y ejecutará en primer lugar admincuestionario.php que será el encargado de decidir que se realiza y que se muestra según las variables que vengan informadas.

Funciones y atributos de las clases:

dbAccess.php

Atributos

dbAbstract: Objeto de tipo dbAbstract

Funciones

|  |  |
| --- | --- |
| dbAccess() | |
| Datos de entrada |  |
| Datos de salida |  |
| Descripción | Crea una instancia del objeto de la clase dbAbstracty lo guarda en el atributo dbAbstract |

|  |  |
| --- | --- |
| crearConexion() | |
| Datos de entrada |  |
| Datos de salida |  |
| Descripción | Crea una conexión a la base de datos. Llama al método crearConexion(servidor,usuario,clave,dbname) de dbAbstract |

|  |  |
| --- | --- |
| getIdPreguntas() | |
| Datos de entrada | * dificultad (String) |
| Datos de salida | * idPreguntas (Array) |
| Descripción | Hace una consulta a la base de datos por dificultad y almacena en un array todas los id de las preguntas. |

|  |  |
| --- | --- |
| getPreguntas() | |
| Datos de entrada | * dificultad (String) |
| Datos de salida | * preguntas (array) |
| Descripción | Realiza una consulta a la base de datos de las preguntas que sean de la dificultad especificada.  Crea un objeto pregunta por cada una y las incluye en un array que será el que devuelva. |

|  |  |
| --- | --- |
| addPreguntas() | |
| Datos de entrada | * pregunta (Objeto de tipo pregunta) |
| Datos de salida | * mensaje (booleano) |
| Descripción | Inserta una pregunta y devuelve si se ha realizado o no. |
| getIdPregunta() | |
| Datos de entrada |  |
| Datos de salida | * id(número de id de la pregunta) |
| Descripción | Comprueba cual es el último id en uso y nos da el siguiente. |

|  |  |
| --- | --- |
| borrarPregunta() | |
| Datos de entrada | * id(número de id de la pregunta) |
| Datos de salida | * mensaje (booleano) |
| Descripción | Borra la pregunta que tenga el id que le enviamos y devuelve si se ha realizado o no. |

|  |  |
| --- | --- |
| editarPregunta() | |
| Datos de entrada | * pregunta (Objeto de tipo pregunta) |
| Datos de salida | * mensaje (booleano) |
| Descripción | Edita la pregunta enviada y devuelve si se ha realizado o no. |

dbAbstract.php

Atributos:

Db: contendrá un objeto de tipo mysqli (conexión a bd)

|  |  |
| --- | --- |
| crearConexion() | |
| Datos de entrada | * servidor(String) * usuario(String) * clave(String) * dbname(String) |
| Datos de salida |  |
| Descripción | Contiene el método para crear la conexión de la base de datos por los parámetros de entrada. |

|  |  |
| --- | --- |
| cerrarConexion() | |
| Datos de entrada |  |
| Datos de salida |  |
| Descripción | Cierra la conexión a la base de datos. |

|  |  |
| --- | --- |
| consulta() | |
| Datos de entrada | * sql (String) |
| Datos de salida | * query (consulta sql) |
| Descripción | Realiza una consulta a la base de datos. |

|  |  |
| --- | --- |
| getResultado() | |
| Datos de entrada | * resultado (consulta sql) |
| Datos de salida | * resultado (array asociativo) |
| Descripción | Obtiene una fila de resultado como un array asociativo |

pregunta.php

Atributos:

- enunciado: Se guarda el enunciado de la pregunta.

- r1: Esta será siempre la respuesta correcta.

- r2: Respuesta alternativa.

- r3: Respuesta alternativa.

- foto: nombre del archivo de la imagen.

-dificultad: fácil, normal, difícil.

|  |  |
| --- | --- |
| function\_\_construct | |
| Datos de entrada | * id * e * res1 * res2 * res3 * foto * dificultad |
| Datos de salida |  |
| Descripción | Construye un objeto pregunta con los valores de entrada. |

vistaAcceso.php

|  |  |
| --- | --- |
| mostrar() | |
| Datos de entrada |  |
| Datos de salida | * Datos del formulario |
| Descripción | La pantalla consiste en dos capas una para el mensaje de bienvenida y la otra con un formulario donde elegimos la dificultad. |

vistaPreguntas.php

|  |  |
| --- | --- |
| mostrar() | |
| Datos de entrada | * Preguntas (array de preguntas) * Dificultad |
| Datos de salida | - Datos del formulario |
| Descripción | La pantalla consiste en una capa superior que contiene dos botones, uno para insertar preguntas que hará visible otra con un formulario y otro botón para volver a la pantalla anterior.  Y seguido se genera una capa por cada pregunta que contiene un formulario con los datos actuales y botones de editar y borrar. |

vistaMensaje.php

|  |  |
| --- | --- |
| mostrar() | |
| Datos de entrada | * mensaje(resultado de la operación) * Dificultad |
| Datos de salida | - Datos del formulario |
| Descripción | La pantalla simplemente muestra el resultado de cualquier operación de inserción, edición o borrado y nos devuelve a la consulta de preguntas de la que veníamos. |

Admincuestionarip.php

Gestiona el flujo principal comprobando si viene algún formulario informado realizando en ese caso mostrar preguntas, inserción, edición o borrado y en caso de que no venga ninguno se mostrará la pantalla de acceso.